

Subject: ETS MPS/Aura Engineering Release 3.5 Delivery
Date: Fri, 25 Apr 2003 10:43:18 -0400
From: Ernest Quintin
To: Willie N Fuller

Willie,

We are pleased to deliver Release 3.5 of the ETS Multimode Portable Simulator (MPS) for Aura. This engineering release delivery contains code corrections to the Scenario module to fix problems discovered during testing of Release 3.4. These problems were recorded in Discrepancy Report ETS0475. The release also includes changes to answer Change Request ETS0474, Synchronize ETSF and MPS time, as well as major enhancements to the Log module, the Event Message disk log, and a number of requests made by ETSF and IPG personnel. Complete descriptions of the changes and enhancements are contained in the attachments.

There are seven attachments to this letter.

Attachment A describes the capabilities included in this release.
Attachment B describes installation instructions for this release.
Attachment C describes special operating instructions for this release.
Attachment D contains the resolved DR descriptions
Attachment E contains the system limitations.
Attachment F contains an updated release history summary matrix.
Attachment G contains an updated Mission Systems Configuration Management (MSCM) form.

The updated software executable modules are being delivered on CD-ROM. Two copies of the CD will be given to Guy Cordier, who will forward one copy to Raytheon at Denver and will use the other for installation on the MPS simulator PCs in Building 32.

The updated software is also being installed on the serial card-equipped PCs in the Bldg 25 Simulations Operations Center, in the event that any of those units are needed to support upcoming Aura data flows.

The System User's Guide is being updated to include the new capabilities.

If you have any questions about this delivery, please do not hesitate to contact me or Estelle Noone.

Ernest Quintin
301-805-3649

CC: wasuncio@eoc.ecs.nasa.gov, jcarlson@pop400.gsfc.nasa.gov,
jcifuent@pop500.gsfc.nasa.gov, gcordier@rattler-e.gsfc.nasa.gov,
elizabeth.ferrara@gd-ns.com, ggarner@rattler-e.gsfc.nasa.gov,
jguida@pop500.gsfc.nasa.gov, tpha@raytheon.com, arhabeger@west.raytheon.com,
Jim King <jking@toronto.eoc.ecs.nasa.gov>, kcklem@west.raytheon.com,
klavery@pop500.gsfc.nasa.gov, rmesserly@mail.gblt.averstar.com,
Robert Patterer <Robert.J.Patterer.1@gsfc.nasa.gov>,
dramey@pop400.gsfc.nasa.gov, diana.sexton@averstar.com,
Candace Shoemaker <Candace.M.Shoemaker.1@gsfc.nasa.gov>,
eshurie@sindbad.gsfc.nasa.gov, "Matt McCoy" <mmccoy@csc.com>,
"Estelle Noone" <enoone@csc.com>, "John Kim" <jkim@csc.com>.

jteter@eoc.ecs.nasa.gov, jtouchst@eoc.ecs.nasa.gov, eweidner@qssmeds.com,
awells@titan.com, william.wingood@titan.com,
"Pat Burrows" <pburrows@csc.com>, "Jean Carlson" <jcarlson@csc.com>,
Rhonda.Winters@titan.com

Attachment A – Summary of Operational Changes

Operational Capabilities of MPS/Aura Release 3.5

New or modified capabilities with this release are noted in **Bold**.

Telemetry:

- Transmit telemetry in IP or Serial (clock/data) mode
- Pack telemetry packets and CLCWs into CADUs when in Serial mode
- Generate one stream of CADUs when in Serial mode
- Generate one stream of telemetry formatted as EDUs when in IP mode
- Start or stop one telemetry stream
- Generate telemetry packets from information contained in the PDB
- Maintain telemetry nodes from information contained in the PDB
- Populate telemetry packets with data values from information contained in the PDB
- Generate correct secondary headers for SC, GIRD, and SUROM-TIE (no secondary header) telemetry packets using information from the PDB
- Generate instrument telemetry packets using secondary key information from the PDB
- Display EDU data when in IP mode
- Display CADU data when in Serial mode
- Set values into telemetry points by mnemonic
- Display telemetry node values by mnemonic
- Convert telemetry values to Engineering Units (EU) for display using information from the PDB
- Accept operator-entered telemetry values in EU and convert to Raw Counts for inclusion in telemetry packets
- **Create a checkpoint file of all telemetry mnemonic values at any time under user control.**
- **Maintain consistency between “mom-kid” telemetry points**
- Reset packet count for the telemetry stream
- Static packet data can be overwritten (by byte location) and by modification of telemetry mnemonic
- Incrementing packet sequence counters per APID
- Generation of individual APIDs can be inhibited
- Telemetry logs will be created (viewable by offline utility)
- Packet Headers and Packet Data are updated
- Packet data can be shown in hexadecimal or octal format and addressed in hexadecimal or decimal form
- Packet Sequence Counters can be reset
- Packet Sequence Counters can be modified
- Packet Version field can be modified
- Packet APID field can be modified
- Packet Type field can be modified

- Packet Secondary Header Flag field can be modified
- Packet Length field can be modified
- CCSDS Unsegmented TimeCode (CUC) can be modified
- Packet rate may be controlled
- CLCW transmitted via EDUs when in IP mode
- IP packets are transmitted with variable lengths
- CLCW can be overridden by the operator
- Transmission of CLCW can be inhibited when in IP mode
- Scenario file (script) capability to set telemetry nodes and buffers
- Set telemetry data values in response to spacecraft commands received (end-item verification)
- Set initial telemetry data values at initialization
- Allow simultaneous display and set of multiple telemetry container items via GUI screens
- Simulate spacecraft memory dumps
- Use the PDB telemetry state text file to locate end-item verifier values
- Maintain and update telemetry data values in APID 1000
- Telemetry parameters may be set and viewed by Parameter ID
- CLCW Transmit Start and Stop is coupled to H/K Telemetry Start and Stop
- Telemetry values may be set using simple expressions
- Telemetry values may be set using trigonometric expressions
- Telemetry values may be set using Boolean expressions
- Telemetry values may be set to other telemetry mnemonic values
- Telemetry values may be saved in intermediate variables for later use
- TES Segmented Packets are emulated
- CLCW Transmit rate may be set by the operator
- Telemetry data values are validated for fit into packet space
- Current enable status and transmit rate for all APIDs is viewable via status display
- vcProcessor module discards VC63 VCDUs when creating files for playback
- The PDB Red/Yellow Limits file is used to refine initial telemetry values.
- Signed telemetry data values are validated as one's and two's complement integers upon user input, as appropriate.
- Displays of telemetry and command container item names may be saved and restored.
- The VCDU Sequence Counter field occupies 32 bits in APID 1000.
- Direct ingest of telemetry-related PDB flat files
- Interface with a 1553 Bus. Transmit telemetry packets over the 1553 Bus.
- Accept telemetry and CLCW packets from an external source in IP mode
- Update telemetry parameter values to reflect data received from the external source
- Update CLCW field values to reflect data received from the external source.
- Forward, via IP interface, the telemetry and CLCW packets received from the external source.
- Modify telemetry parameter values and CLCW field values in externally received packets prior to re-transmission, in response to operator directive.

- Accept CADUs from an external source in serial mode
- Extract telemetry packets and CLCWs from externally received CADUs
- Pack externally received telemetry packets and CLCWs into CADUs and forward via serial interface
- Preserve time in Secondary Headers of externally received telemetry packets OR replace with MPS-generated time.
- Support for operator entry of 1750A telemetry values as Engineering Units.

Command:

- Identify commands using information from the PDB
- Display event messages with command mnemonics and submnemonics
- Set telemetry points in response to commands received (end-item verification) using information from the PDB
- Recognize spacecraft Command Loads
- Display Command Load data
- Copy Command Load data to a Memory Dump buffer
- Inhibit the Command Load data copy facility via operator directive
- Validate checksums of received Command Loads
- Ingest type AD, BC, and BD commands
- Display Total CLTUs count
- Reset Total CLTUs count
- Display Rejected CLTUs count
- Reset Rejected CLTUs count
- Display Instrument commands count
- Reset Instrument commands count
- Display Spacecraft commands count
- Reset Spacecraft commands count
- Display BC commands count
- Reset BC commands count
- Display BD commands count
- Display current Spacecraft CLCW
- Update Spacecraft and instrument CLCW
- Display current Instrument CLCW
- Validate commands based on individual, all, or none of the following validation criteria: CLTU Start and Tail Sequences, BCH Error Code, Transfer Frame Header Fields, FARM (Valid Frame Sequence), User Command Packet Header
- Generate event messages based on ingest
- Log raw commands (viewable by offline utility)
- Display raw command in hexadecimal or octal format addressed in either hexadecimal or decimal fashion
- Display command packet headers for instrument commands
- Display command packet headers for spacecraft commands
- Update command accepted and rejected counters in telemetry

- Command submnemonics are saved in container items and may be viewed after command receipt
- Expected Spacecraft ID changed to CC Hex
- TES and OMI segmented commands are recognized.
- The Function Code is used to identify HIRDLS commands.
- The two's complement checksum of instrument commands is validated.
- Direct ingest of command-related PDB flat files
- Enable and disable automatic setting of end-item verifier telemetry points for commands received, in response to operator directive.
- Interface with a 1553 Bus. Receive command packets from the 1553 Bus.

Time:

- Maintain and update SC time (GIRD)
- Maintain and update GMT time
- Synchronize SC and GMT times
- Set SC time to time in externally received telemetry packets.
- Set SC time to time broadcast via the ETSF 1553 bus.

General:

- Control all simulator module functions via scenario scripts
- Selection of scenario scripts may be via operator type-in or via a file selection browse window
- Start scenario scripts in response to commands received
- Start a scenario script from a scenario script
- Execute multiple scenario scripts simultaneously
- Provide operator control of multiple scenario scripts started by the operator
- Save the last 10 operator directives
- Allow editing of saved operator directives before re-execution
- EDOS Service Header (ESH) fields may be viewed
- ESH field contents may be modified by the operator
- Validation of Command Data Block (CDB) header fields of commands received
- Modification of expected values of CDB header fields
- All viewable buffers may be displayed
- Addition, deletion, and modification of command end-item verifiers via SQL scripts
- Logs of commands received or telemetry transmitted may be retransmitted via IP output or Serial output
- Expected Spacecraft ID may be modified in EOSGS module
- CLCW ESH field contents may be modified by the operator
- Event messages to the screen may be inhibited or enabled by severity (color)

- Scenario scripts may contain IF-then-ELSE-ENDIF and WHILE-ENDWHILE conditional execution directives
- The Scenario module may interface with multiple modules
- **The user may create a disk file giving the names of all running Scenario scripts at any time.**
- **The user may kill any or all running Scenario script(s) at any time.**
- Intermediate variables A – Z permit saving values as real numbers – extended to all modules that accept directives
- Intermediate variables Aq – Zq permit saving values as long integers – extended to all modules that accept directives
- **CREATE and DELETE directives permit the user to create temporary variables of Real type.**
- The Serial Output module can accept directives from the operator or via a scenario script.
- The Event Message window has been separated from the project window and has been made resizable.
- Receipt of CADUs from a serial interface and extraction of telemetry packets is via the EOSXtract module.
- Interface with the 1553 Bus is via the E1553Bus module.
- **Improved telemetry and command logging capability.**
- **Improved Event Message logging capability.**

Attachment B – Installation Instructions for MPS/Aura Release 3.5

This attachment contains the instructions for installing the PDB files and the MPS/Aura Release 3.5 Server and Client. The information presented in this attachment is divided into three major sections. The first section contains abbreviated installation instructions, the second contains a summary of the installation changes, and the third section contains detailed instructions for performing initial and subsequent installations.

The abbreviated installation instructions assume that the user will install the recommended version of the Java Runtime Engine, JRE 1.2, Build 14. If the user desires to remain with the currently installed version of Java, consult the directions for modifying the Client startup batch file in the detailed instructions section.

The information presented in this attachment has been checked for accuracy by the independent test team.

B-1: Abbreviated Installation Instructions

These instructions are intended for the experienced user.

1. Install the Java Runtime Engine, Build 14, by selecting the file **jre-1_2_2_014-windows-i586.exe** in the root folder of the CD and following the prompts.
IMPORTANT: When prompted for an installation folder, modify the path to **C:\jre1.2.2**. See the detailed instructions for more information.
2. Install the MPS/Aura Release 3.5 Client software by executing the **Setup.exe** program in the Client folder of the CD.
3. Install the MPS/Aura Release 3.5 Server software by executing the **Setup.exe** program in the Server folder of the CD.
4. If not previously done, create a folder under **D:\mps_pdb\AuraPDBs** to hold the Aura PDB source files. Copy the Aura PDB source files into this new folder. Twelve files are needed. See the list in Paragraph B-3.3 for the files to be copied.
5. When initializing the MPS/Aura simulator for the first time, all Projects needed must be built and saved.

B-2: Summary of changes

Patch Build 14 of Version 1.2.2 of the Java Runtime Engine (JRE) is being included with this delivery. Testing has shown that Sun Microsystems has fixed many of the resource leaks that were a problem with earlier versions of the JRE.

B-3: Detailed Installation Instructions

This is the complete procedure for performing an initial or subsequent installation of the MPS/Aura simulator Release 3.5, and associated software, data files, and COTS programs on a PC.

Materials Needed:

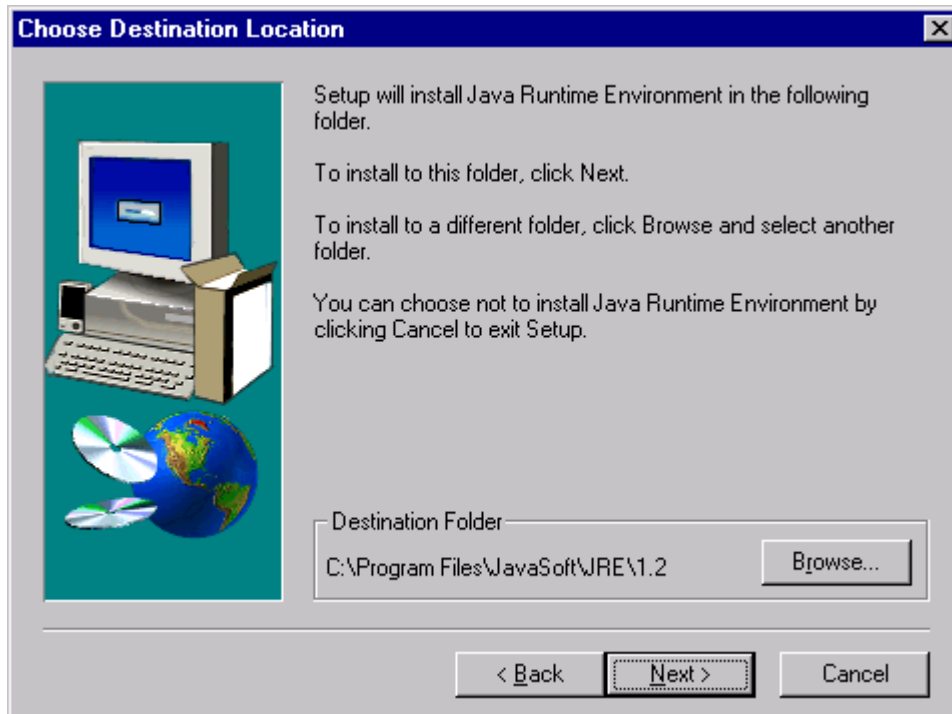
- One or more versions of the Aura Project Data Base (PDB)
- The CD containing the MPS/Aura Release 3.5 software

B-3.1: Java Runtime Engine Installation

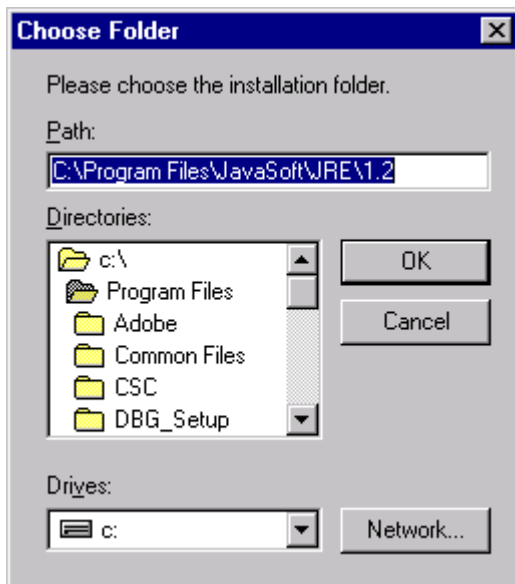
Patch Build 14 of Version 1.2.2 of the JRE is included on the delivery CD. It is recommended that this version of Java be installed. If this version of Java is already installed then skip to Paragraph B-3.2.

1. Insert the CD containing the MPS/Aura Release 3.5 into the CD drive and navigate to it using either Windows Explorer or My Computer.
2. Double-click on the file named **jre-1_2_2_014-windows-i586.exe** in the root folder. This will cause the Java Runtime Engine to be installed. When responding to the installation prompts, set the installation folder to **C:\jre1.2.2** by modifying the installation path as shown in the following pictures. This is necessary because the DOS program that starts the Client software cannot parse spaces in the path.

When the **“Choose Destination Location”** window appears, click on the Browse button.



The result will be the following **“Choose Folder”** browse window:



Modify the path in the **Choose Folder** browse window to be **C:\jre1.2.2** as shown in the following picture. Select **OK**, then select **NEXT** from the **Choose Destination Location** window.

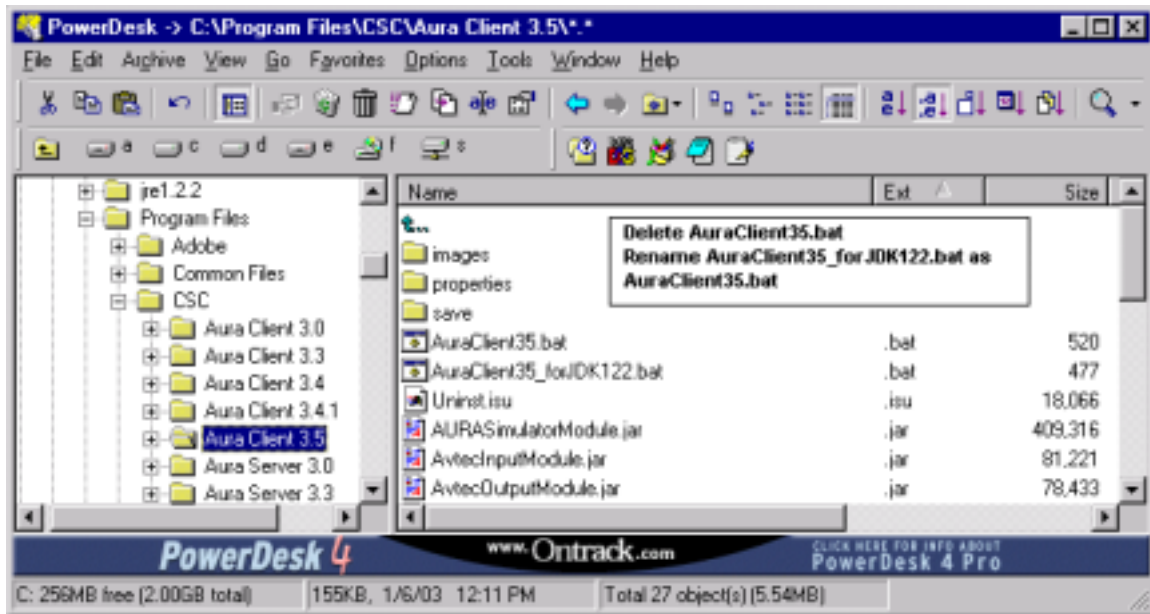


B-3.2: Installation of the Aura Server and Client software

The steps in this paragraph cause the MPS/Aura Client and Server software to be installed on the PC.

1. Insert the delivery media into the appropriate drive.
2. To install the Aura Client:
 - a) On the desktop, click on the Start button, and then select Run from the resulting menu.
 - b) When the Run window appears select the Browse... button.
 - c) From the Browse Window, select the Removable drive that contains the installation CD.
 - d) Click on the Client folder.
 - e) From within the Client folder, double click on the **Setup.exe** filename.
 - f) A window with the title "Run Window" will appear. Click on the Okay button to proceed to the next step.
 - g) The screen will be filled with an Aura Client background and a smaller window with the title "Welcome to Aura Client 3.5" will appear. Click on the Next button to proceed to the next step.
 - h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
 - i) After all of the files are copied, a window with the title "Setup Complete" will appear. Click on the Finish button to end.
 - j) An Aura Client icon will now be installed on the desktop.

3. If the newer version of the Java JRE was not installed, you must modify the Client startup batch file. Follow these steps.
 - a. Using either Windows Explorer or My Computer, navigate to the **C:\Program Files\CSC\Aura Client 3.5** folder.
 - b. As indicated in the following picture, delete the original **AuraClient35.bat** file.
 - c. Rename **AuraClient35_forJDK122.bat** as **AuraClient35.bat**.



4. To install the Aura Server:
 - a) On the desktop, click on the Start button, and then select Run from the resulting menu.
 - b) When the Run window appears select the Browse... button.
 - c) From the Browse Window, select the Removable drive that contains the installation CD.
 - d) Click on the Server folder.
 - e) From within the Server folder, double click on the **Setup.exe** filename.
 - f) A window with the title "Run Window" will appear. Click on the Okay button to proceed to the next step.
 - g) The screen will then be filled with an Aura Server background and a window with the title of "Welcome to Aura Server 3.5" will appear. Click the Next button to proceed.
 - h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
 - i) Next a window will show the completion status as the files are copied. When the copying is complete click on the Finish button to finish the installation.
 - j) An Aura Server icon will be installed on the desktop.

B-3.3: PDB Download

The next step is to copy the PDB onto the hard drive. You will need at least one version of the Aura PDB. The following PDB flat files are needed, where *xxxxxx* corresponds to the version portion of the filename:

```
cmd_desc_XXXXXX.pdb  
cmd_fixdata_XXXXXX.pdb  
cmd_parm_XXXXXX.pdb  
cmd_vardata_XXXXXX.pdb  
cmd_verify_XXXXXX.pdb  
t1m_calcurve_XXXXXX.pdb  
t1m_desc_XXXXXX.pdb  
t1m_dstate_XXXXXX.pdb  
t1m_packet_XXXXXX.pdb  
t1m_parm_XXXXXX.pdb  
t1m_polyconv_XXXXXX.pdb  
t1m_rylim_XXXXXX.pdb
```

Add a folder to your chosen directory structure to hold the source files of the Aura PDB.

Copy the desired version of the PDB into the folder just created. If desired, more than one version of the PDB may be copied. Be sure to copy each version into its own folder.

Attachment C - Special Operating Instructions

This attachment contains new special operating instructions for MPS/Aura Release 3.5. The information presented in this attachment has been checked for accuracy by the independent test team.

A User's Guide is being updated to include the information presented in this section. When complete, the User's Guide will be available from the ETS home page at <http://romulus.gsfc.nasa.gov/ETS>.

Name Changes to Executable Files

Updates in the Configuration Management repository have required name changes to the ETS-specific modules. The new and old module names are shown in the following table. The modules operate exactly as before. The picture following the next paragraph shows a typical IP-mode Project.

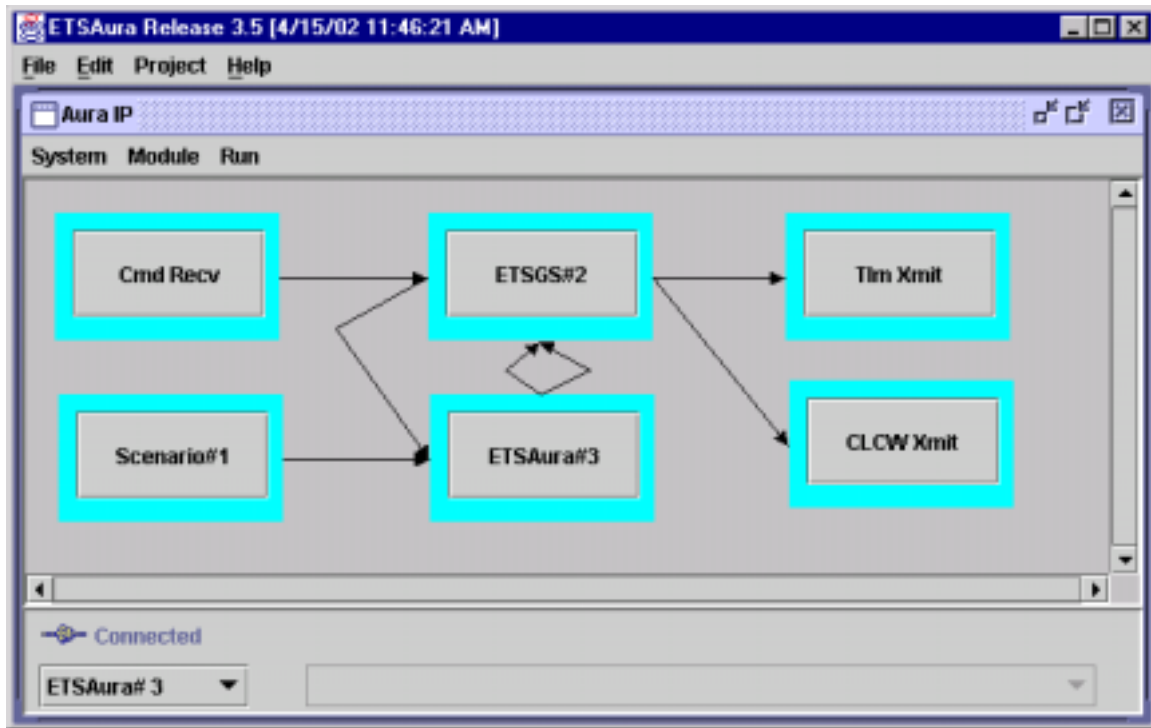
Old Module Name	New Module Name
EOSGS	ETSGS
SCAura	ETSAura
EOSXtract	ETSXtract
E1553Bus	ETS1553Bus

Renaming of Projects and Modules

Another enhancement recently made available in the SIMSS baseline is the capability to rename Projects and Modules. This capability is illustrated in the following picture, which illustrates a typical IP-mode project where many of the modules have been given descriptive names.

To change the Project name first Stop and Unlock the Project, if it is running. Select **Rename** from the **System** menu. Enter the desired Project name into the pop-up window. The new Project name will appear at the top left-hand corner of the Project and Event Message windows.

To change a Module name first Stop and Unlock the Project, if it is running. Position the mouse over the desired module and single-click. Select **Rename** from the drop-down menu. Enter the desired Module name into the pop-up window. The new Module name will appear at the top left-hand corner of every window associated with that module. The new name will also appear in the drop-down module selection menu at the bottom of the Project screen.



CREATE and DELETE directives

The CREATE and DELETE directives have been added to enhance the usability of scenario scripts. The CREATE directive will create a variable of type Real with the name given by the user. The DELETE directive will delete a variable. The usage rules are as follows:

- The user may create as many temporary variables as are needed.
- Variables may be created via the directive entry line or within scenario scripts.
- The variables are local to the module in which they are created. Only the Scenario module may access variables created in another module.
- The created variables remain in existence until either the module, or Project, is removed from the GUI, or the simulator is exited.
- The variable name syntax is not case sensitive. Eg. “XYZ” is identical to “xyz”, “Xyz”, “XyZ”. Etc.
- Variable names may be up to 64 characters long.
- While the creation logic makes no restriction on the characters making up a variable name, avoid using special characters (+ (plus), - (minus), (space), / (division), etc.) because the resulting variable will be interpreted as an expression by some parts of the system. Using underscores is a good way to enhance readability.
- If the variable being created already exists, a warning message will be sent to the Event log. The simulator will use the existing variable.

- The Create and Delete keywords are not case sensitive.
- The following examples illustrate the CREATE and DELETE directives.

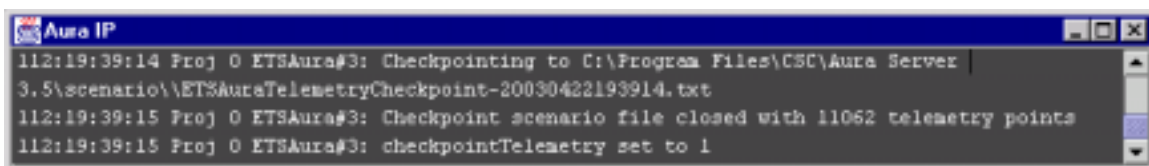
```
Create Outer_loop_counter
Create Save_GNC_SS_ACTVEPH_value
.
.
.
Delete Outer_loop_counter
```

Telemetry Checkpoint Capability

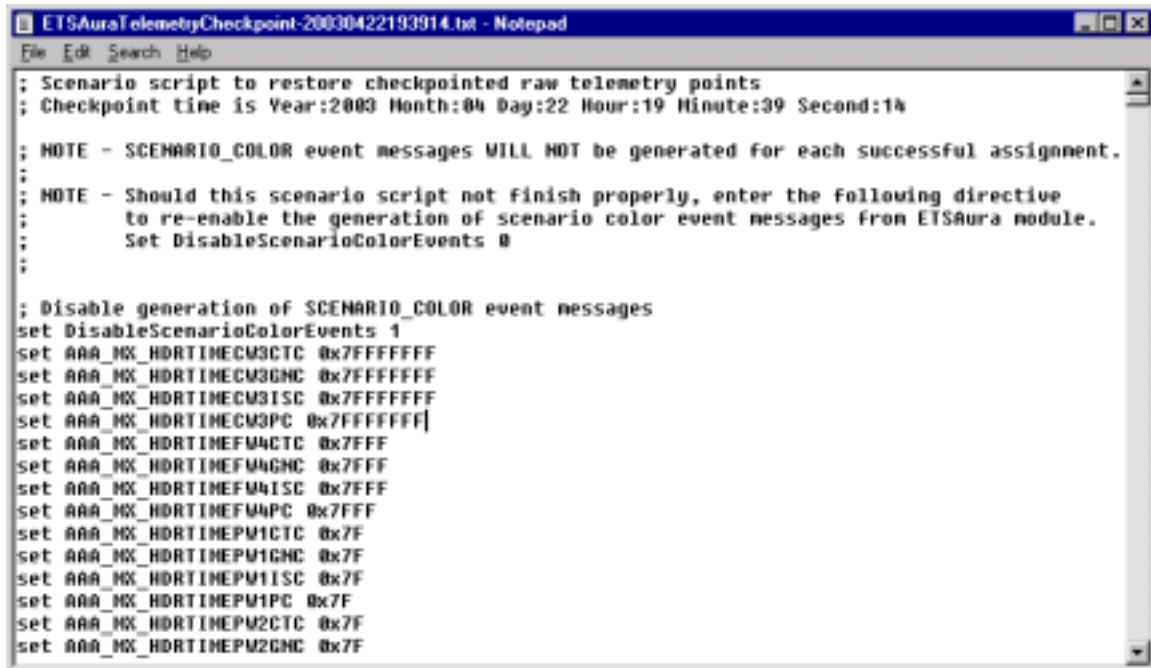
A telemetry value checkpoint capability has been added to the ETSAura module. The facility is manual rather than automatic so that the user may create a checkpoint file whenever desired, and so that the file creation will not interfere with normal simulator operations. Issue the following directive to create the checkpoint file.

Set checkpointTelemetry 1

When the directive is issued, a file with scenario script format will be created in the Scenario folder. The filename contains the year, month, day, hour, minute, and second when the creation process started. The file consists of set directives with the current value of every defined telemetry point. A NEW FILE WILL BE CREATED EVERY TIME THE DIRECTIVE IS ISSUED. Event messages will be generated when the checkpoint file creation begins and ends, as shown in the following picture.



The following is a fragment of the checkpoint scenario file, showing the entry format. To restore the simulator to the checkpointed state, select the checkpoint file as a scenario script and run it. Since the checkpoint file is a scenario script, the value of any telemetry point may be edited prior to reading the file back in.



```
ETSAuraTelemetryCheckpoint-20000422193914.txt - Notepad
File Edit Search Help

; Scenario script to restore checkpointed raw telemetry points
; Checkpoint time is Year:2000 Month:04 Day:22 Hour:19 Minute:39 Second:14

; NOTE - SCENARIO_COLOR event messages WILL NOT be generated for each successful assignment.
;
; NOTE - Should this scenario script not finish properly, enter the following directive
;        to re-enable the generation of scenario color event messages from ETSAura module.
;        Set DisableScenarioColorEvents 0
;

; Disable generation of SCENARIO_COLOR event messages
set DisableScenarioColorEvents 1
set AAA_HX_HORTIMECM3CTC 0x7FFFFFFF
set AAA_HX_HORTIMECM3GNC 0x7FFFFFFF
set AAA_HX_HORTIMECM3ISC 0x7FFFFFFF
set AAA_HX_HORTIMECM3PC 0x7FFFFFFF
set AAA_HX_HORTIMEFM4CTC 0x7FFF
set AAA_HX_HORTIMEFM4GNC 0x7FFF
set AAA_HX_HORTIMEFM4ISC 0x7FFF
set AAA_HX_HORTIMEFM4PC 0x7FFF
set AAA_HX_HORTIMEPM1CTC 0x7F
set AAA_HX_HORTIMEPM1GNC 0x7F
set AAA_HX_HORTIMEPM1ISC 0x7F
set AAA_HX_HORTIMEPM1PC 0x7F
set AAA_HX_HORTIMEPM2CTC 0x7F
set AAA_HX_HORTIMEPM2GNC 0x7F
```

Synchronization of Coincident Telemetry Points

Code has been added to the ETSAura module to keep the values of coincident telemetry points, colloquially known as mom-kid telemetry, in sync. This capability is automatic and requires no special action on the part of the operator.

Because of this capability, telemetry initial value processing had to be changed slightly. All one-bit telemetry points are set to zero but their value may change if they have a “mom” and it is set to non-zero. Also, the values obtained from the Red-Yellow Limits Project Database (PDB) file are checked to see if they fit within the parameter space. If the value does not fit it is not used.

ETS1553Bus module changes

The E1553Bus module has been modified so that the simulated spacecraft time will be synchronized with the time signal from the EOS Training Simulator Facility (ETSF) 1553 Bus.

Scenario module changes

MPS/Aura Release 3.5 contains the first of a number of changes to make the Scenario module a more useful tool. See the Limitations attachment for more information. The changes in this release are as follows:

Client:

The Client GUI has not really been changed. However all variables have been renamed to match their slot position. In the previous version, the variables for slot number 1 had no number and were labeled scenarioFilename, scenarioStatus, etc. Slot 2 items were scenario1Filename, scenario1Status, etc. Now, slot 1 variables are named scenario1Filename, scenario1Status, etc., slot 2 variables are named scenario2Filename, etc., and so on through slot 5.

Server:

1. Every scenario script run causes an execution thread to be generated. Each thread is now assigned an ID number between 1 and 65535. The five GUI scenario threads are numbered 1-5. Beyond that, every new script is assigned a new thread number. Thread numbers become free when the script ends and are re-used when the ID wraps around. Note that the five GUI scenario threads always exist. This means that, after wraparound, the next thread number will be the first free number after five.
2. All event messages from the Scenario module now include the scenario ID and line number, if applicable.
3. All scenario threads execute concurrently. The scenario property, ConcurrentScenario, is no longer used. The code to implement serial execution of scenario scripts has been removed.
4. When referencing a scenario filename from within a scenario, or in the Command-Scenario.txt trigger file, relative pathnames may be used to specify the file location.
5. The “killthread” variable has been modified to use the scenario ID numbers. When the killthread variable is set to 0, all generated threads are deleted and all GUI threads are stopped. If the killthread variable is set to any other number, an attempt will be made to kill or stop the single scenario thread with that ID. If no script with the given ID number is running, the directive has no effect.

Eg. Set killthread 95

will kill the script with ID number 95 if it is running.

6. A “scriptStatus” variable has been created in the Scenario module. When this variable is set to any value, a file containing current scenario thread information is written into the default scenario directory. The default scenario directory is defined by the ScenarioPath variable. The file that is generated has the following name format: *Scenario#<x>ScriptStatus<y>*, where <x> is the scenario module number and <y> is the value of the scriptStatus variable.

Eg. Assuming the Scenario module is module #3 then

set scriptStatus 1

will result in the file *Scenario#3ScriptStatus1.txt* being created in the *Scenario* folder. If a file with that name already exists it will be overwritten. An example of the file contents follows.

```
; Scenario script information
; Year:2003 Month:04 Day:20 Hour:19 Minute:37 Second:32
;
; Script Parent Run Line IfCnt While Filename
; 1 GUI No 231 0 0 C:\Program Files\CSC\Aura Server
3.5\scenario\Start 16k Telemetry - Aura PDB 112102-01.txt
; 2 GUI No 0 0 0
; 3 GUI No 146 0 0 C:\Program Files\CSC\Aura Server
3.5\scenario\IPG Scenario Scripts\OMI\OMIS_constant_updates.txt
; 4 GUI Yes 6 0 0 C:\Program Files\CSC\Aura Server
3.5\scenario\RunLongWait.txt
; 5 GUI No 0 0 0
; 6 4 Yes 56 0 0 C:\Program Files\CSC\Aura Server
3.5\scenario\cdh_hir_longWait.txt
; 7 4 Yes 10 0 0 C:\Program Files\CSC\Aura Server
3.5\scenario\Aura_polynomial_conversion_longWait.txt
```

The script status file contains the following information.

- Script ID. The first five ID numbers correspond to the five GUI-controlled scripts.
 - Parent. Whether the script was started via the GUI, spawned by another script, or in response to command received.
 - Current execution status.
 - Current line number.
 - Depth within nested IF statements.
 - Depth within nested WHILE statements.
 - Script name, including its complete path. For the GUI-controlled threads the most recently executed script will be displayed.
7. If a scenario script is running a WHILE loop, and no SLEEP statements were executed within the most recent loop iteration, an automatic SLEEP of one second will be performed at the bottom of the loop.
8. A bug was fixed in the processing of Nested IF statements within While Loops. This bug caused the ENDIF statement for the first IF statement to be ignored. After the first execution of the loop, every additional IF test was considered a new level of nesting. This bug caused many additional resources to be consumed by

these threads, the wrong scenario statements to be executed, and the invalid “missing ENDIF” event messages.

9. The following script debugging aids have been added. While the debugging aids may be enabled for any scenario before or during execution, they will probably be most helpful if enabled for a scenario being run from the GUI before the script is started.

NOTE: Due to the large volume of additional event messages generated when the following flags are enabled, additional delays are added to the script processing by the Scenario module to help prevent Event Log overflow.

- Each scenario thread has been given a directive flag variable of the format “Scenario<ID>dirmmsg”, where ID is the script ID number. For scripts started from the GUI the ID will be 1, 2, 3, 4, or 5, as appropriate. When this flag is set, the scenario will echo the script directive lines to the Event window. When the debug flag is not set (see below), these event messages are the Scenario color of dark gray. If the debug flag is also non-zero, these event messages are the Information color of white.
 - Each scenario thread has been given a debug flag variable of the format “Scenario<ID>debug”, where ID is the script ID number. When this flag is set, the scenario will send debugging information to the Event window. These event messages are the Debug color of light gray.
10. The Scenario script syntax has been enhanced to allow comments on the same line as directives. All characters after the ‘;’ are ignored.
 11. The “Stop Scenario <filename>” command has been implemented for use within Scenario scripts. <filename> may include a full or relative path or the filename only. All scripts that match <filename> will be stopped.

Log Module Changes

The Log module has been extensively modified, both Client and Server. To describe the changed module, the Log Module chapter from the draft MPS/Aura Release 3.5 User’s Guide is included in its entirety.

Log Module

Log-1.0 Overview

The Log module writes the data it receives to a specified log file.

Viewing or printing of log files may be accomplished by means of an external program. The software used must be capable of displaying binary data in an ASCII representation. The shareware utility, Hexedit, which is available over the Internet from Alexander Reidel Informations-Systeme, is one such program.

Log-2.0 Inputs

Ch	Data expected	Validation performed	Processing performed
1	Bytes	None	If logging is enabled, received log data is written into the log file.

Log-3.0 Outputs

The Log module does not have any output channels.

Log-4.0 Container Items

The Log module is capable of receiving directives from the operator from the directive line or from a scenario file. Container item names are not case-sensitive. Values for string type variables should not include quote characters. When configuring the Log module from a scenario, set the “OpenNewLog” container item to one as the last parameter to force the opening of a log file with the previously specified parameters.

Name	Type	Project Saved?	Description
DESCFLAG	Number	Saved	Optional description flag (1=description field is included in generated filename, 0=field not used)
DESCFLD	String	Saved	Configured description field
DIRNAME	String	Saved	Configured log directory
EXTFLAG	Number	Saved	Optional extension flag (1=file extension is included in generated filename, 0=field not used)
EXTFLD	String	Saved	Configured file extension field
FILECOUNT	Number	Not saved	Current full file count
FILESIZE	Number	Not saved	Current log file size
FIXEDFLAG	Number	Saved	Fixed format flag (1=every log entry will be the same fixed size, 0=log entries have variable lengths)
FIXEDSIZE	Number	Saved	Log item size in bytes when the Fixed format is selected
FORMATFLAG	Number	Saved	Optional format flag (1=format characters are included in generated filename, 0=field not used) When the format flag is set, the following letters are inserted in the

			generated filename: H if log headers are used N if log headers are not used V if variable length entries are written F# if fixed length entries are written # is the fixed record size in bytes
HEADERFLAG	Number	Saved	Log header flag (1=log with header, 0=no header)
ITEMSLOGGED	Number	Not Saved	Number of records logged
ITEMSTRUNCATED	Number	Not Saved	Number of truncated records
LOGGINGENABLED	Number	Saved	Logging enabled flag (1=enabled, 0=disabled)
MAXFILECOUNT	Number	Saved	Maximum full file count limit (0=no limit, 2 thru 999=file limit)
MAXFILESIZE	Number	Saved	Maximum size of each log file in bytes.
OPENNEWLOG	Number	Not Saved	Open new log request (0 = open if parameters have changed, 1 = always open new log file)

Log-5.0 Displays

To access displays for the Log module, click in the center of the module in the project window. The module pop-up menu will appear. The “Remove” option can be used during project design to remove this module. The “Configure” option may be used prior to running the project and at run-time. The Run-time option is available only when the project is running.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

Log-5.1 Configuration

Selecting the “Configure” pop-up menu option produces a display similar to the following screen. Default values are shown for all parameters. The Log module will generate a name for the log file that includes the project number, module name, module number and a time field. This prevents filename collisions between any two Log modules on the same server PC in the same or different projects. By including a time stamp field, there should be no accidental overwriting of recently logged data.

Log #0 Configuration

☐ Logging Enabled

Log Directory: ...

Generated log file name includes:

☒ Description Field:

☒ Log Format Specification

☒ File Type Extension:

Log Format:

☒ Log With Header

☒ Variable Length Size in bytes:

File Controls:

Max File Size (bytes):

Max Full File Limit:

Field	Default Value	Default Description
Logging Enabled	Disabled	This field MUST be toggled to enabled to start data logging.
Log Directory	"Log" directory below installed release directory	Sample display shows a development directory.
Description Field Box	Selected	The associated description field will be added to the generated log file name.
Description Field String	Empty	No default description field
Log Format Specification	Selected	Letters will be included in the generated filename to indicate whether the file records contain Log Headers and Variable or Fixed length entries.
File Type Extension Box	Selected	The associated extension field will be added to the end of the generated filename.
File Type Extension String	"log"	Associated file type extension field
Log With Header Box	Selected	Data will be logged with log headers. Log headers record the length of the entry and the logged time.
Variable Length Box	Selected	Variable length records will be written into the log.

Size in bytes	256	This field is not applicable for the variable length format.
Maximum File Size	1000000 bytes	When the current log file reaches this size, it is closed and a new log file is opened.
Maximum Full File Count	0 indicating no limit	By default, there is no limit on the number of log files that may be created..

Log-5.1.1 Logging Enabled Flag

The Logging Enabled flag allows the operator to control when a Log module logs data. Since this flag is initially disabled, several Log modules may be added to a project without being configured. Later during runtime, these modules may be configured. At the very least this flag **MUST** be changed to enabled to start logging data.

NOTE

When a Log module is paused, stopped or logging disabled, the Log module box on the project window has a red border. When a Log module is running and logging is enabled, the module's box has a green border.

Log-5.1.2 Log Directory

The default location for the log files is in the "Log" directory below the directory where the server executables were installed. The operator may define a different directory location. A file browse button is provided.

Log-5.1.3 Description Fields

When the Description Field Box is checked, the characters in the Description Field are included in the generated log filename.

Log-5.1.4 Log Format Specification Field

When this option is selected, characters are added to the generated log filename that indicate whether the data was logged with headers or not and whether the log entries are fixed or variable in length.

Knowing how the data was logged makes a log file easier to use with other SIMSS utilities. Log files with headers may be retransmitted using the same timing intervals as originally logged. Variable length format logs must have log headers to be usable. The fixed length record size is needed to process fixed format files if log headers were not used.

When the Log Format Specification option is specified the following letters are added to the file name.

Format Characters	Format Description
H	Indicates logged With Log Headers
N	Indicates logged with No Log Headers
V	Indicates Variable length records
Fn	Indicates Fixed length records of n bytes. For example F256 indicates Fixed format with 256 byte records.

Log 5.1.5 File Type Extension Fields

When the File Type Extension Box is checked, the characters in the File Type Extension String are included at the end of the generated log filename. It is not necessary to precede this field with a period character.

Log 5.1.6 Log With Header Format

When the Log With Header box has been checked, all log records will begin with a log header of ten bytes. This header consists of eight bytes of system time followed by two bytes of data length. The following example shows two CLCW packets. The header in each packet has been circled for clarity. When the Log With Header box is not checked, all records will be written without a header.

Address	Hex Data
00000000	01 BF 85 59 4F 6D 6A A0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
00000015	80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 01 01 00 00 00
0000002A	01 BF 85 59 4F 7F C0 E0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
0000003F	80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 02 01 04 00 00

Log-5.1.7 Variable Length Format

When the Variable Length box has been checked, varying length records are written to the log file. No truncation or padding of received data is done. When this field is checked, the fixed size field is desensitized. When the Variable Length box is not checked, the fixed length format is selected and the fixed size field is enabled for input.

Log-5.1.8 Size in bytes

The Size field defines the Fixed format length of the data record to be written to the log file. A size that is shorter than the actual data received will result in truncation of the data that is logged. A size that is longer than the actual data received will be zero filled to the specified size and then written to the log file.

Log 5.1.9 Maximum File Size (bytes)

The Maximum File Size in bytes limits the size of individual log files. When the next item to log would exceed the maximum size, the current log file is closed and a new log file is opened. The new log file will use the currently configured file naming options but

with a newer timestamp field. The item is then logged in the new log file. This parameter may not be set smaller than 12,000 bytes.

Log 5.1.10 Maximum Full File Count

The Maximum Full File Count limits the total number of “full” log files generated by a Log module. Each time the Maximum File Size is reached, the current log file is closed and the full path name of the closed file is saved. When the number of “full” log file names reaches the maximum file count, the oldest log file is deleted. When the maximum full file count is set to zero, there is no limit on the number of files created by the Log module. When this parameter is non-zero, it must be a value between 2 and 999.

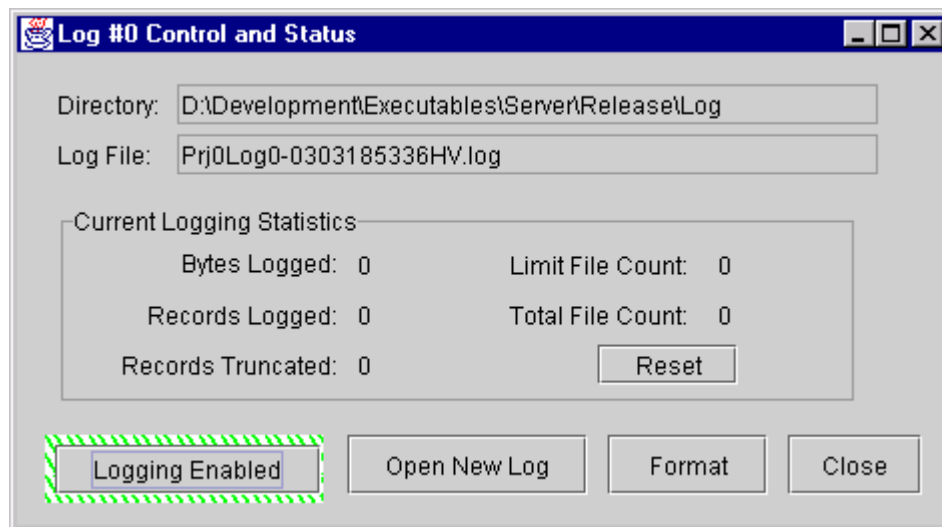
Log-5.2 Run-time

Clicking the “Run-time” option of the module pop-up menu produces a Run-time menu.

Run-time Menu Item	Description
Configure	Request Configuration Display (see Log-5.1.1)
Show Status	Request Control and Status Display
Pause (or Resume)	Pause (or Resume) the module
Stop (or Restart)	Stop (or Restart) the module

Log-5.2.1 Control and Status Display

When the “Show Status” option is selected from the Run-time Menu, a screen like the following is displayed. These fields do not accept data entry.



Status Fields	Description
Directory	This field shows the directory for the current log file.
Log File	This field shows the current log file name.
Bytes Logged	Total bytes written to the current log file. This count includes log headers when headers are being used.
Records Logged	Number of records written to the current log file.

Records Truncated	Number of records truncated because the data was larger than the configured fixed record size. This is an indication that all data is not being captured in the log. This statistic is only applicable to the Fixed log format.
Limit File Count	Number of “full” files that count toward the maximum full file limit. Full log files are automatically closed by the software prior to exceeding the maximum file size. This number only increments when there is a maximum full file limit.
Total File Count	Total number of non-empty files that have been created and closed by this Log module. This number includes files closed due to the maximum file size limit and files closed by the Open New Log button. This number is reduced when file limit processing causes file deletions.

Log 5.2.1.1 Reset Button

The Records Truncated count shows the number of records that were truncated prior to being logged. The first time a truncation occurs, a warning event message is also generated. The reset button is only associated with this field. Each time this count is reset, the next truncation produces another warning event message.

Log 5.2.1.2 Logging Enabled/Disabled Button

The Logging status button is used to toggle the logging enabled flag. When logging is enabled, this button is labeled “Logging Enabled” and has a green border. When logging is disabled, this button is labeled “Logging Disabled” and has a red border.

NOTE

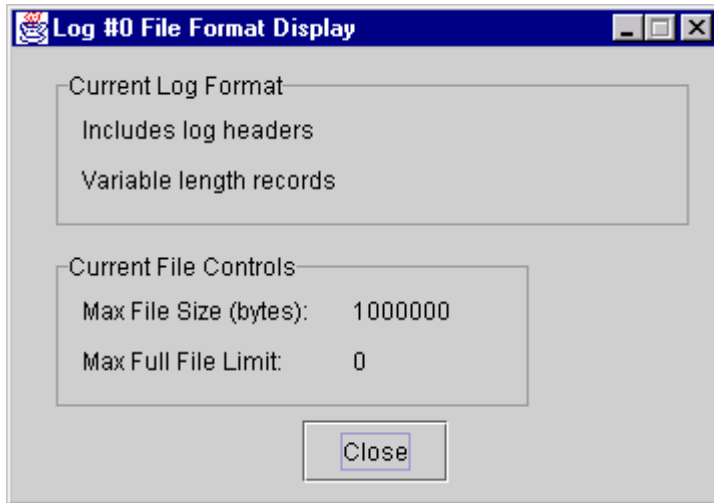
The current logging status affects the color of the Log module box in the project window. When a Log module is paused, stopped or logging disabled, the Log module box has a red border. When the Log module is running and logging is enabled, the module’s box has a green border.

Log 5.2.1.3 Open New Log Button

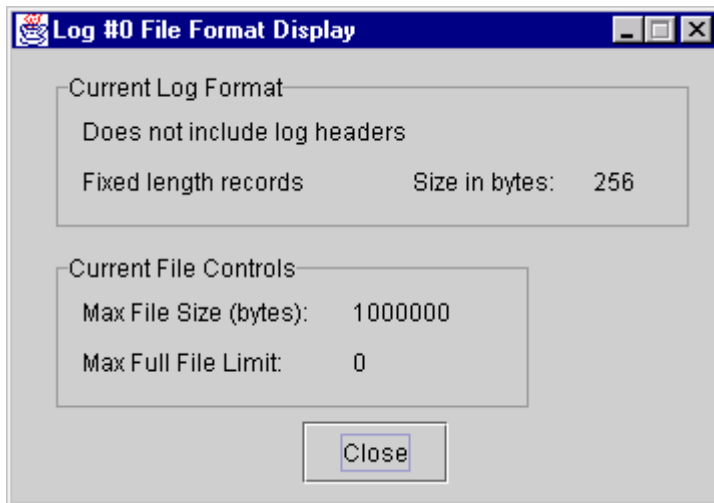
The Open New Log button is used to force the closing of the current log file and open a new log file. The new log file will use the latest configured log parameters. Use of this button has no immediate effect on maximum file limit processing. When a maximum full file limit has been specified, the names of ALL non-empty closed log files are saved. These partial files are automatically deleted only after the maximum amount of new data has been logged.

Log 5.2.1.4 Format Button

The Format button on the Control and Status Display is used to request a display of the current logging format and file limits. Some sample Format displays are shown below. These fields do not accept data entry.



When a fixed format is used, this display also shows the fixed record size.



Log-5.2.2 Pause/Resume

Select "Pause" from the Run-time Menu to pause the Log module's processing. The module's border changes to red. The text of this menu option changes to "Resume". While the Log Module is paused, it does not receive or log data. After a Log module has been paused, select "Resume" from the Run-time Menu to restore processing. If logging is enabled when the module is resumed, the border color changes to green.

Log-5.2.3 Stop/Restart

Select "Stop" from the Run-time Menu to stop the Log module's processing. The module's border changes to red. The text of this menu option changes to "Restart". After a Log module has been stopped, select "Restart" from the Run-time Menu to restart the module's execution. If logging is enabled when the module is restarted, the border color changes to green.

Log-5.3 About

Selecting the “About” option from the module pop-up menu produces a display that lists the module’s number of input links, number of output links, whether directives are allowed, names of authors and the version number.

Log-6.0 Special Operating Instructions

There are no special operating instructions for this release.

Attachment D – Resolved Discrepancy Reports

In addition to the new capabilities, the following Discrepancy Report (DR) and Change Request (CR) have been closed by and are being delivered with MPS/Aura Release 3.5. The DR and CR are listed in the table below, which provides the DR/CR Number, Status, Severity, and a short description. A full description of each DR/CR follows the summary table. Complete information on all DRs/CRs may be accessed via the Internet at address <http://edosultra30.gsfc.nasa.gov/ddts/>

Summary of Closed Discrepancy Reports

Critical (Severity 1)	Urgent (Severity 2)	Routine (Severity 3)	Change Requests	Total
0	1	0	1	2

Status Definitions

N – New	A – Assigned Analysis	R – Analysis Entered
V – Assigned Verification	T – Tested	C – Closed
W – Withdrawn	P – Postponed	X – Duplicate

ETS No.	SMO No.	Type	Severity	Description
ETS0474	SMOdr20076	CR	2	Need to synchronize ETSF and MPS 1553 Bus PC time.
ETS0475	SMOdr20078	DR	2	Scenario script causes MPS crash if stopped.

DR: SMOdr20076 (ETS0474) Related NCR: Submitted: 030324
Status: NEW Class: ETS

Title: Need to synchronize ETSF and MPS 1553 Bus PC time

SUBMITTAL INFORMATION

Project: ETS
DR Type: Change Request
Rel/Ver: 3.4
Subsystem: Aura
Module: Simulator
Affected-Requirement:
Test Phase: in-field use
Severity: 2
Date found: 030313
Location: GSFC
Submitter: Ernest Quintin
Organization: ETS Dev Group
Phone number: 301-805-3649
Email: equintin@csc.com

***** Problem (Added 030324 by equintin) *****
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

This problem was recognized during the Aura SCIF #3 dry run.

ETSF is given a spacecraft start time during initialization.
MPS derives its spacecraft time from current GMT. Thus, MPS time
does not usually sync with ETSF time. The result is that spacecraft
time displayed at EMOS jumps back and forth. Automatic synchronization
of MPS time with ETSF is very desirable.

The current workaround is to manually synchronize MPS time with ETSF.

***** History *****

batchbug 030324 211849 Submitted to ETS by equintin
batchbug 030324 211849 Enclosure "Problem" added by equintin

DR: SMOdr20078 (ETS0475) Related NCR: Submitted: 030324
Status: NEW Class: ETS

Title: Scenario script causes MPS crash if stopped

SUBMITTAL INFORMATION

Project: ETS
DR Type: Problem
Rel/Ver: 3.4
Subsystem: Aura
Module: Simulator
Affected-Requirement:
Test Phase: in-field use
Severity: 2
Date found: 030313
Location: GSFC
Submitter: Ernest Quintin
Organization: ETS Dev Group
Phone number: 301-805-3649
Email: equintin@csc.com

***** Problem (Added 030324 by equintin) *****
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

The scenario script OMIS_Constant_Updates.txt causes an MPS crash
if stopped by the operator rather than being allowed to run to
completion. It exits properly if allowed to run to completion.

This was observed during the Aura SCIF #3 dry run. The script was
provided to MPS developers by the IPG team.

***** History *****

batchbug 030324 212712 Submitted to ETS by equintin
batchbug 030324 212712 Enclosure "Problem" added by equintin

Attachment E – System Limitations

E.1 MPS/Aura Release 3.5 Limitations

The following limitations apply to MPS/Aura Release 3.5. Some of these are Discrepancy Reports (DRs) against SIMSS baseline products and have been recorded in their DR repository.

Problem Description	Workaround
The Scenario module will occasionally crash the simulator if it encounters multiple nested IF and WHILE statements. An MPS DR will be written.	There is no specific workaround. Users must attempt to devise workarounds specific to their script execution. The problem is being worked.
The Scenario module will incorrectly interpret an expression as a variable if spaces are omitted. E.g. “ABC < 3” works as expected but “ABC<3” will be misinterpreted.	Ensure that all the elements of arithmetic expressions are separated by spaces.
The Log Module file browser appears to have deficiencies in that it will not always switch directories when requested. The problem is being worked.	Take the default folder name when configuring the Log Module.
The Save Project (Extended) and Restore From (Extended) options are intended for another application where a remote server runs simultaneously with the local application. The options are included with MPS/Aura so that only one version of the NeTTCore code needs to be maintained.	Avoid use of the Save Project (Extended) and Restore From (Extended) options.
The Generic Container Buffer display is limited to 1400 bytes of data (= 700 words, or 350 double words). A request for more data than that will result in a display of 1400 bytes of information. <i>This is SIMSS Defect # 102.</i>	To view data that is beyond byte 1400 of the buffer, set the offset to 1400, or as required to view the data.

Problem Description	Workaround
<p>Certain APIDs are included in the PDB tlm_packet file without any Interval or Slot number information. MPS supplies a default Interval of one second and a default Slot number of zero. However, the packet timing appears to be less accurate than for those APIDs that have at least one Interval field filled in the tlm_packet file. More investigation is necessary before writing this as an MPS DR.</p>	<p>Explicitly supply an Interval when enabling an APID that has no non-zero Interval in the tlm_packet file.</p>
<p>The E1553Bus module will crash MPS if it is invoked on a PC that does not have a 1553 Bus interface board installed.</p>	<p>Avoid adding the E1553Bus module to any Project if the PC does not have a 1553 Bus interface board.</p>
<p>When converting 1750A parameter values to Engineering Unit Floating Point format, MPS rounds the result to six decimal places.</p>	<p>External calculations may be used to check the accuracy of the raw values transmitted in telemetry.</p>

Attachment F - Release History Summary Matrix

Attached is the MPS/Aura simulator release history summary matrix, updated to reflect the MPS/Aura Release 3.5 delivery. Modules inherited from the SIMSS baseline have the SIMSS Release Number, while the MPS/Aura modules ETSGS, ETSXtract, ETS1553Bus, and ETSAura have their current Release Number.

Release History Summary Matrix

System: **MPS/Aura**

Release Number		1.0	2.0	3.0 Beta	3.0	3.1	3.2	3.3	3.4	3.5				
Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02	8/9/02	8/23/02	10/25/02	1/10/03	4/25/03				
Configuration Item	CI No.													
Core (Client)	1.1	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
Core (Server)	1.2	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
ETSAura (Client) Formerly SCAURA	1.3	1.0	2.0	3.0	3.0	3.1	3.2	3.3	3.4	3.5				
ETSAura (Server) Formerly SCAURA	1.4	1.0	2.0	3.0	3.0	3.1	3.2	3.3	3.4	3.5				
ETSGS (Client) Formerly EOSGS	1.5	1.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	3.5				
ETSGS (Server) Formerly EOSGS	1.6	1.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	3.5				
IP Input (Client)	1.7	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
IP Input (Server)	1.8	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
IP Output (Client)	1.9	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
IP Output (Server)	2.0	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
Logging (Client)	2.1	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				
Logging (Server)	2.2	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0				

Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02	8/9/02	8/23/02	10/25/02	1/10/03	4/25/03					
Configuration Item	CI No.														
Scenario (Client)	2.3	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
Scenario (Server)	2.4	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
Serial Input (Client)	2.5	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
Serial Input (Server)	2.6	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
Serial Output (Client)	2.7	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
Serial Output (Server)	2.8	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
TxFile (Client)	2.9	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
TxFile (Server)	3.0	4.0	4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
vcProcessor (Client) ¹	3.1		4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
vcProcessor (Server) ¹	3.2		4.1	5.0	6.0	6.0	6.0	7.0	7.0	8.0					
ETSXtract (Client) ² Formerly EOSXtract	3.3							1.0	1.0	1.0					
ETSXtract (Server) ² Formerly EOSXtract	3.4							1.0	1.0	1.0					
ETS1553Bus (Client) ² Formerly E1553	3.5							1.0	2.0	3.0					
ETS1553Bus (Server) ² Formerly E1553	3.6							1.0	2.0	3.0					

¹ Added with Release 2.0

² Added with Release 3.3

Attachment G — Mission Systems Configuration Management Form

This attachment contains the completed Mission Systems Configuration Management (MSCM) form for the delivery of MPS/Aura Release 3.5.

Mission Systems Configuration Management Form

<u>1. ORIGINATOR</u> Estelle Noone	<u>2. ORGANIZATION</u> CSC	<u>3. PHONE</u> 301-805-3653	<u>4. E-MAIL ADDRESS</u> enoone@csc.com
<u>5. ELEMENT</u> ETS (MPS/Aura)		<u>6. INSTALLATION PRIORITY</u> Routine	<u>7. TRACKING NUMBER</u> (Assigned by CM Office)
<u>8. SOURCE CHANGE REQUEST(S):</u> ETS delivery of MPS for EOS Aura (MPS/Aura)		<u>9. APPROVALS</u> <div style="display: flex; justify-content: space-between;"> <div>Element Manager</div> <div>_____</div> <div>____/____/____</div> </div> <div style="display: flex; justify-content: space-between;"> <div>Flight Ops Director</div> <div>_____</div> <div>____/____/____</div> </div> <div style="display: flex; justify-content: space-between;"> <div>Operations Manager</div> <div>_____</div> <div>____/____/____</div> </div>	
<u>10. DELIVERED SYSTEM</u> (Check all that apply)			
	Name	Version	Media Identification
<input type="checkbox"/> Hardware	_____	_____	_____
<input checked="" type="checkbox"/> Software	MPS/Aura	R3.5	CD-ROM
<input type="checkbox"/> Database	_____	_____	_____
<input checked="" type="checkbox"/> Documentation:			
	MPS/Aura delivery package	N/A	via email
	MPS/Aura Release 3.5 User's Guide	R3.5	http://esdis-it.gsfc.nasa.gov/ETS/etsdoc.html
	_____	_____	_____
<input type="checkbox"/> Other	_____	_____	_____
<u>11. CHANGE DESCRIPTION</u> Release 3.5 of MPS/Aura _____ _____ _____			
<u>12. ATTACHMENT(S):</u> Check if YES <input checked="" type="checkbox"/> Description: MPS/Aura Release 3.5 delivery package (cover letter with attachments) dated 4/25/03 _____ _____			
<u>13. CM OFFICE USE</u>			
	Location (Bldg/Room)	Slot location(s)	
Hardware	_____/____	_____	
Media	_____/____	_____	
Documentation	_____/____	_____	
Installation date	____/____/____	CM Office Signature _____	

Form MSCM (970327)

Attachment G — Mission Systems Configuration Management Form

This attachment contains the completed Mission Systems Configuration Management (MSCM) form for the delivery of MPS/Aura Release 3.5.

Mission Systems Configuration Management Form

<u>1. ORIGINATOR</u> Estelle Noone	<u>2. ORGANIZATION</u> CSC	<u>3. PHONE</u> 301-805-3653	<u>4. E-MAIL ADDRESS</u> enoone@csc.com
<u>5. ELEMENT</u> ETS (MPS/Aura)		<u>6. INSTALLATION PRIORITY</u> Routine	<u>7. TRACKING NUMBER</u> (Assigned by CM Office)
<u>8. SOURCE CHANGE REQUEST(S):</u> ETS delivery of MPS for EOS Aura (MPS/Aura)		<u>9. APPROVALS</u> <div style="display: flex; justify-content: space-between;"> <div>Element Manager</div> <div>_____</div> <div>____/____/____</div> </div> <div style="display: flex; justify-content: space-between;"> <div>Flight Ops Director</div> <div>_____</div> <div>____/____/____</div> </div> <div style="display: flex; justify-content: space-between;"> <div>Operations Manager</div> <div>_____</div> <div>____/____/____</div> </div>	
<u>10. DELIVERED SYSTEM</u> (Check all that apply)			
	Name	Version	Media Identification
<input type="checkbox"/> Hardware	_____	_____	_____
<input checked="" type="checkbox"/> Software	MPS/Aura	R3.5	CD-ROM
<input type="checkbox"/> Database	_____	_____	_____
<input checked="" type="checkbox"/> Documentation:			
	MPS/Aura delivery package	N/A	via email
	MPS/Aura Release 3.5 User's Guide	R3.5	http://esdis-it.gsfc.nasa.gov/ETS/etsdoc.html
	_____	_____	_____
<input type="checkbox"/> Other	_____	_____	_____
<u>11. CHANGE DESCRIPTION</u> Release 3.5 of MPS/Aura _____ _____ _____			
<u>12. ATTACHMENT(S):</u> Check if YES <input checked="" type="checkbox"/> Description: MPS/Aura Release 3.5 delivery package (cover letter with attachments) dated 4/25/03 _____ _____			
<u>13. CM OFFICE USE</u>			
	Location (Bldg/Room)	Slot location(s)	
Hardware	_____/____	_____	
Media	_____/____	_____	
Documentation	_____/____	_____	
Installation date	_____/____/____	CM Office Signature _____	

Form MSCM (970327)